Project Report to COMP0028
Video Synopsis in Timelapse

1 Introduction

Video synopsis is one type of time lapse which reduces video's spatio-temporal redundancy. Using video synopsis enables quick content browsing. A dynamic video synopsis will condense most activities in video by showing different actions simultaneously [1]. In practice, such synopsis also been called stroboscopic movie.

In this project, I will show an Matlab implementation for creating video synopsis. After the generic implementation, I will focus on some extension work such as panoramic synopsis and addressing some issues related to generic approach.

2 General Video Synopsis Approach

Figure 1 gives a board Illustration of video synopsis, a synopsis will visualize spatio-temporal-different activities in a single frame.

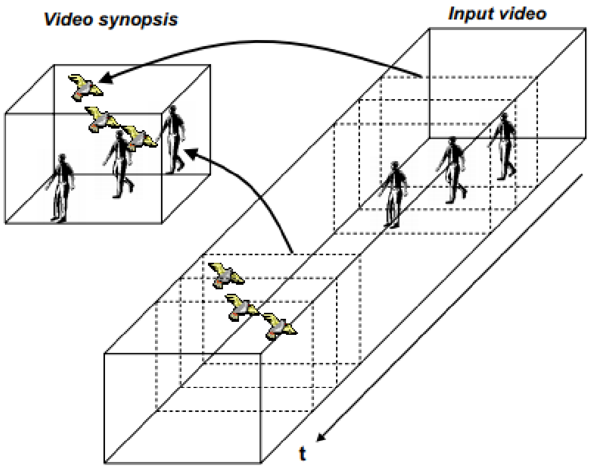


Figure 1: Video Synopsis Illustration[1]

In general, a video synopsis approach can be concluded into two steps: activity detection and frame condensation. Among them, detecting major activity is a challenging and critical part in synopsis task.

2.1 Activity Detection

Activity Detection can be further divided into background and foreground separation. One basic approach for such separation is to use medians of input video. In a still background scene, the pixel value of background is approximately equal to the temporal pixel medians in video. Using Matlab library function `median()`, we can easily achieve the expected result.

Once we have that, we can use image and background difference to get the foreground elements.

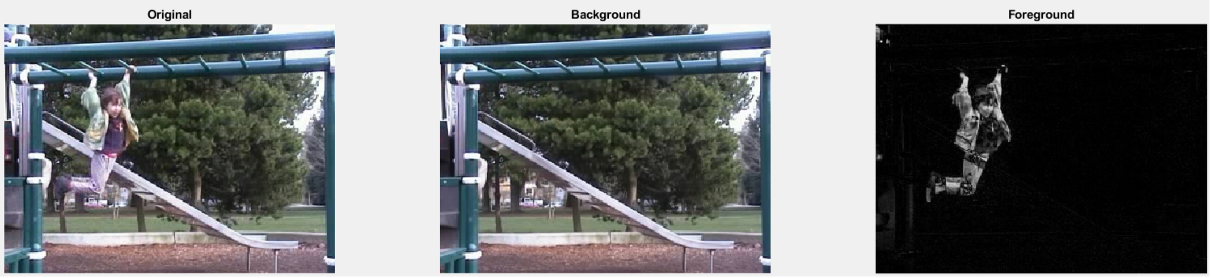


Figure 2: Separation Result

Figure 2 gives the result of introduced separation approach in frame 48. We can see the background is very realistic and the foreground result gives enough contextual information for later process.

With foreground object detected, next thing to do is making a binary mask for those detected object. I start with Otsu's thresholding[2] mechanism(`im2bw()`)to get the raw object binary mask.

The mask will then be processed by a close operation(`imclose()`) for de-noise purpose and a region growing method to maintain a consistent object window.

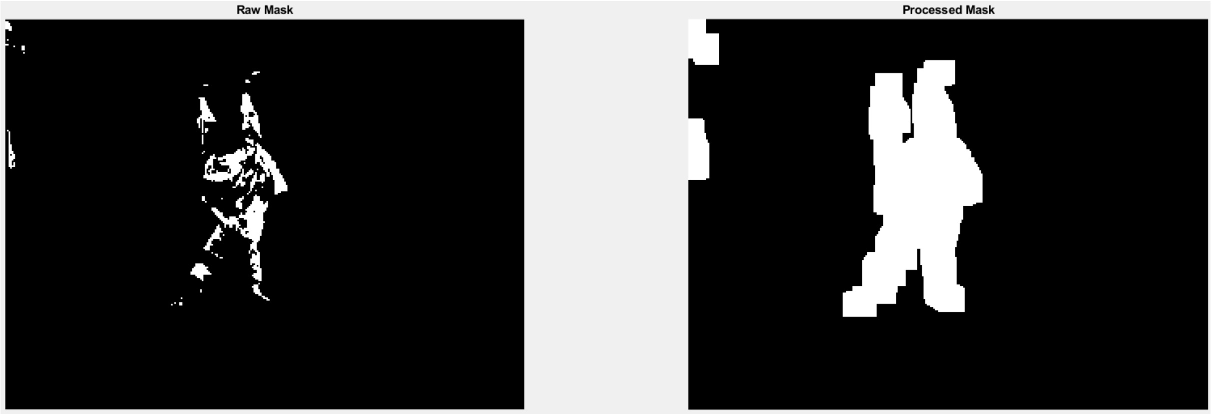


Figure 3: Mask Processing Result

Figure 3 shows the result of my proposed mask processing approach. The mask covers the area of main target which should be considered as a good object mask. One noticeable observation in my mask is it's top left corner, these result is due to a second girl showing up in the frame, our mask successfully detect and encircled her as well.

2.2 Frame Condensation

Condensation requires a simple user interaction by specifying how *condensed* the synopsis are. In my case, I use the number of objects on frame to measure the degree of *condense.*

After we obtained the background image and activity mask, we can proceed to put them together. This process is quite straightforward: divide video into user specified sections, for each section, stick detected object into its spatial location at its corresponded time step into the background.

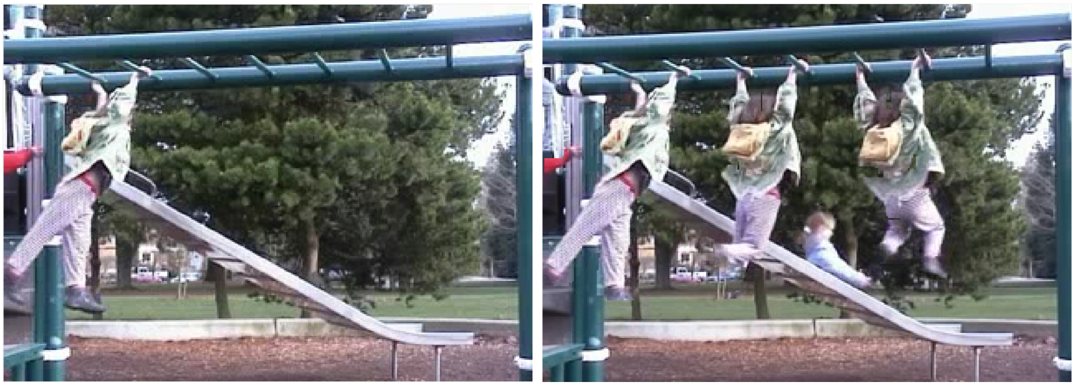


Figure 4: Synopsis Result

According to Figure 4, image on the left is the input frame at $t = 5$. With the same frame step, image on the right gives the synopsis output. As we can seem the result from the figure gives the expected output of video[1] synopsis.

However, at some frame of synopsis we may notice some artefacts, in Figure 5, we can see some holes within the object. These holes can be fixed if I do a finer tuning of close operation as well as more region growing. We may also spot the occlusion in the middle of frame, this violates the original assumption made for video synopsis that the object should not occlude each other. Further advanced algorithm may be necessary to avoid such occlusion but they are beyond the scope of this project.

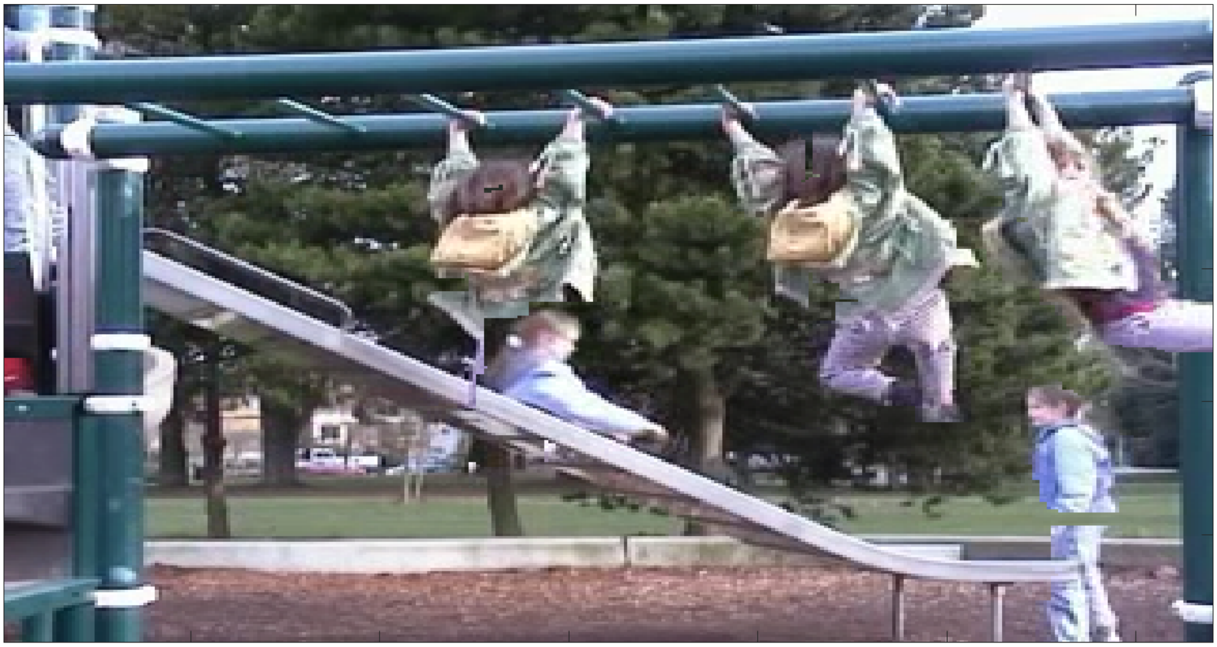


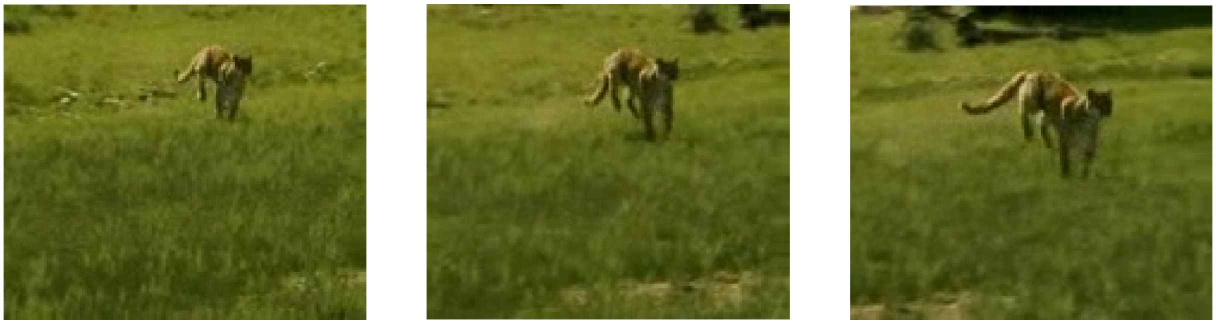
Figure 5: Synopsis Result with Some Artefact

The relevant matlab file is `basicBGseparation`.

3 Video Synopsis over Dynamic Background

In previous section, I proposed a simple video synopsis approach based on still background. In reality, however, many videos are not in a steady background but in a animated background.

[1] This footage comes from http://www.vision.huji.ac.il/video-synopsis/



(a) $t = 10$ (b) $t = 30$ (c) $t = 50$

Figure 6: Input Sequence Example at Different Time steps t

In this section, I will address such issue by proposing a video synopsis approach which uses the idea of panorama.

According to Figure 6, it shows three frames from input footage[2]. As we can see, the background is panning and the object's perspective also changed. Thus, we cannot simply borrow ideo from last section to get background directly.

In this case, we need to track the object as one way for foreground extraction and using panoramic stitching to get the background.

3.1 Object Tracking

The objective of object tracking is to draw a rectangular window that correctly encompasses the object of interest. The success of my proposed object tracking approach is based on the assumption of certain object movement between each consecutive frames. Since our object is changing shape constantly due to it's sprinting posture. To track such object, I compute the optical flow for each frame as a measure of object activity. The calculation of optic flow is based on tool box(*Coarse2FineTwoFrames()*) introduced in coursework2[3].

[2] This footage comes from http://www.vision.huji.ac.il/video-synopsis/

[3] https://people.csail.mit.edu/celiu/OpticalFlow/

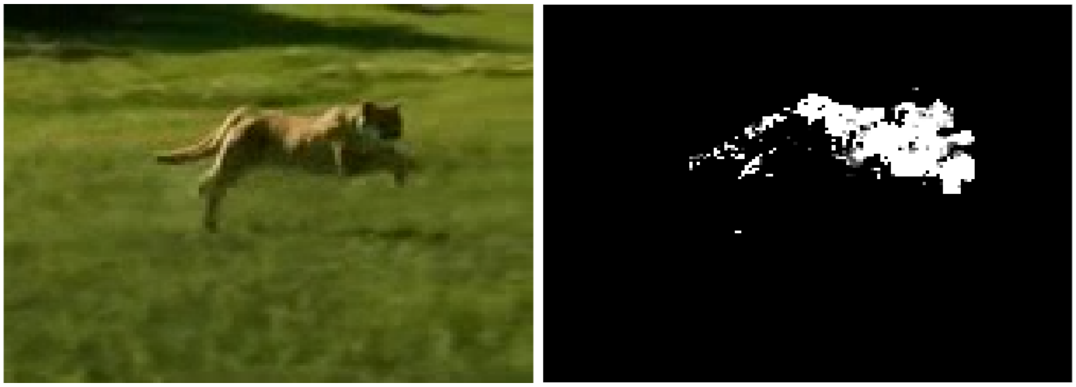


Figure 7: Optic Flow Result

As we can see in Figure 7, the output of optic flow looks similar to difference in median and image as I did in last section. Therefore, I can use same processing procedure to make a object mask. The rectangular window is created based on the largest area in the mask.

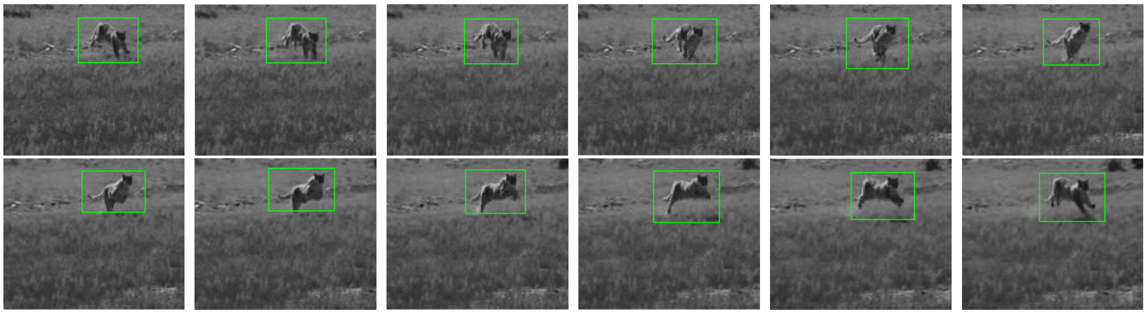


Figure 8: Tracking Result

We can see the tracking result in Figure 8 with consecutive 12 frames. My proposed tracking method is perform well in this task. The tracking result from 80 frames with only 4 frame that failed to detect object and another four frame that misses the tail of tiger. The main cause for tracking failure is the beginning 5 frames are identical with no visual noticeable activity, so optic flow is not suitable in such situation.

3.2 Panorama Stitching

After successfully tracked the object, next thing to do is to remove the tiger in each frame. This can be done by taking tracking window as mask and filling windowed area with median pixel value.

One typical object removal result can be seen in Figure 9. From the figure, we can see the tiger gets removed but left a blurry fillings. Although result is not perfect, it is still enough for making a panorama.

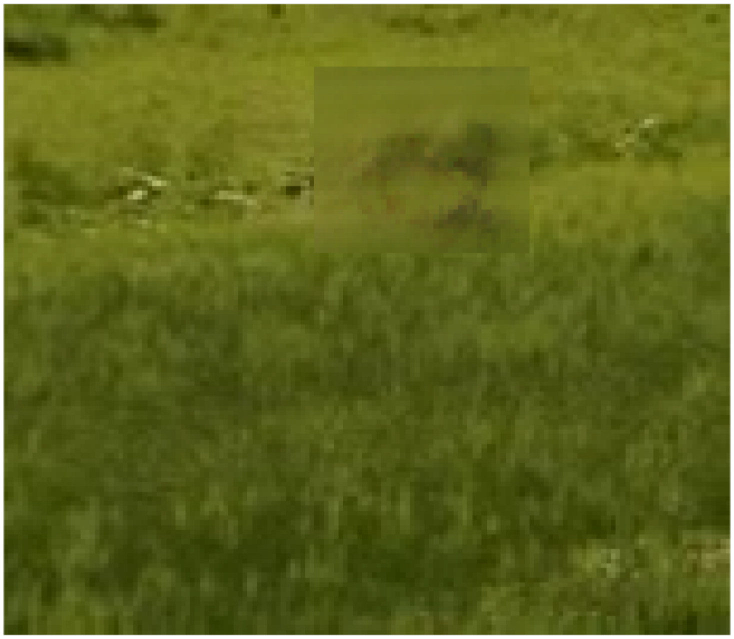


Figure 9: Object Removal Result

Making panoramic is based on finding geometric relationship between each consecutive frames. In my project, I use *homography* as the composite matrix transformation between frames:

$$\mathsf{I}' = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \mathsf{I}$$

Where I and I' stands for a pair of consecutive frames.

In Matlab, a more efficient way for estimating homography is to use SURF feature matching. It involves SURE feature extraction(`extractFeatures()`), feature matching(`matchFeatures()`) and geometric relationship estimation(`estimateGeometricTransform`) between frames.

For arbitrary frame, its homography to others can be expressed by a series of composite homography multiplication start from current frame to target frame. So just pick a reference frame, and compute composite homography between itself and rest of frames, we get the aligned sequence based on reference frame. Then I use `blender` to stitch frames together to form panoramic background.

The stitching result can be seen in Figure 10.

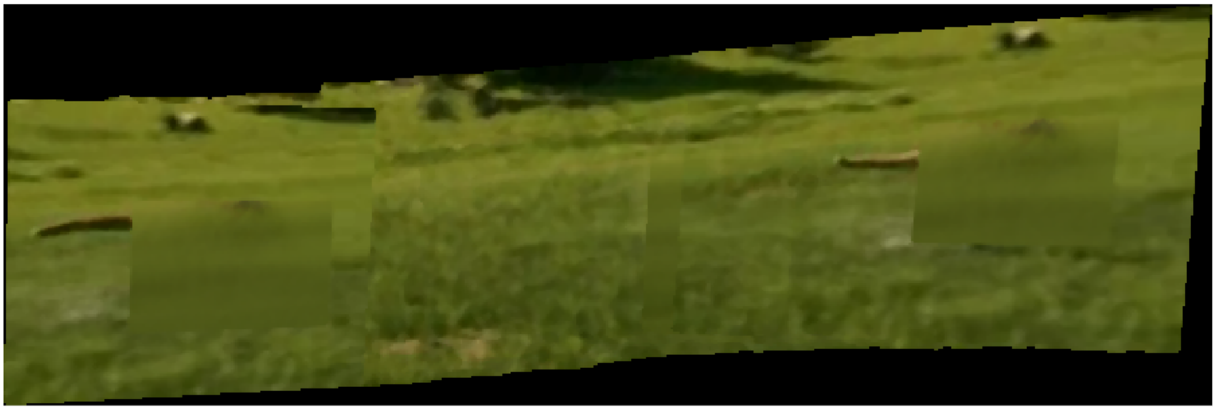


Figure 10: Panorama Result

We indeed get the panorama image, but the quality is not as expected, the main artefact is the tail. Detecting tail can be improved by comparing optic flow between temporal further frames instead of neighbour frame. We can also see the effect of blurry filling is reduced since most of them are overlapped, that is why I don't come up with more realistic method for fillings.

Since we get the background and the object mask, what is left is similar to previous section which is sticking object into the panorama.

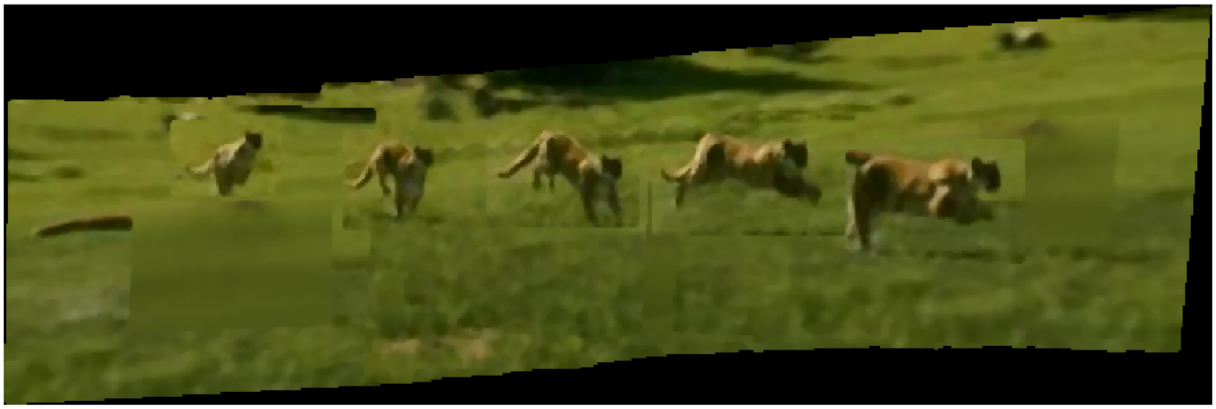


Figure 11: Panorama Synopsis Result

The synopsis result in given in Fiugre 11. We can see some hard rectangular edges for some tigers, this is due to the tracking window mask. To circumvent such effect, I can use object mask I obtained using optic flow directly instead of transferring it to tracking window. Another way for blending tiger into background is to use Poisson Image Editing I implemented for Image Processing coursework. But I think using such method is an overkill and occupies so many computational powers.

The relevant matlab file is `PanoramicSynopsis`.

4 Conclusion

In this project, I provide a video synopsis implementation based on matlab. I first propose a basic approach based on still background. Then I further extend my approach to make synopsis method applicable to dynamic background and forms a panorama. I achieved the expected result for the general approach and a acceptable result in panorama. I also give my evaluation and future improvement works at the end of each section.

References

[1] A. Rav-Acha, Y. Pritch, and S. Peleg, "Making a long video short: Dynamic video synopsis," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*, IEEE.

[2] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, jan 1979.